NASA Technical Memorandum 86304

Communication Protocols for a Fault Tolerant, Integrated Local Area Network for Space Station Applications

Barry D. Meredith

September 1984

NASA
National Aeronautics and
Space Administration
Langley Research Center
Hampton, Virginia 23665

Summary

The evolutionary growth of the Space Station and the diverse activities onboard are expected to require a hierarchy of integrated, local area networks capable of supporting data, voice and video communications. In addition, fault tolerant network operation is necessary to protect communications between critical systems attached to the net and to relieve the valuable human resources onboard Space Station of day-to-day data system repair tasks. An experimental, local area network is being developed which will serve as a testbed for investigating candidate algorithms and technologies for a fault tolerant, integrated network. The establishment of a set of rules or protocols which govern communications on the net is essential to obtain orderly and reliable operation. A hierarchy of protocols for the experimental network is presented and procedures for data and control communications are described.

Introduction

The National Aeronautics and Space Administration is devoting part of its research efforts toward the development of technologies that will support the needs of an orbiting Space Station. One area of involvement for Langley Research Center has to do with the development of data system technologies which fit into the context of Space Station as currently envisioned. The Space Station should be established in earth orbit in the early 1990's and will initially consist of a minimum number of modules or compartments. There would then be a buildup phase where compartments are added and joined as per the functional activities onboard the station. In the mature phase, the Space Station will have reached its maximum structural size (but would allow for the replacement of modules) and will possess its maximum operational capability.

The evolutionary growth of the Space Station, along with the uncertainty of that growth has implications on the design of the data system. Electronic systems also need to be added or taken away to meet changing requirements and to accommodate the continuing advances in electronics technology. One potential solution involves establishing a hierarchy of computer networks for Space Station.[1,2] Each module would contain its own local area network[3,4] (LAN) which would be joined to networks in other modules via gateways or nodes to form a Space Station-wide network. The Space Station net would be able to communicate with earth-based networks over a telecommunications channel producing a global data communications and processing system.

While there are a large variety of local area networks currently available,[3,4] they possess some notable weaknesses in terms of their ability to address the diverse data communication requirements of the Space Station. Commercially available LAN's lack the performance characteristics necessary to accommodate certain high data rate applications (100 megabits/sec.), e.g. the transfer of real-time, video data to and from points in the Space Station. Full motion video displayed at various work stations would augment a workers ability to perform proximity operations onboard the Space Station. These operations might involve a remote process, which is monitored by a video camera, and a person in the loop who is workihg with a mechanical manipulator. In addition to the performance limitations, most existing networks are employed for office automation tasks and provide little (if any) fault tolerance. The ability to detect and recover from network faults is particularly essential when critical systems, such as on-orbit control, are attached to the net and dependent upon it for reliable data communications. It is also desirable to employ fault tolerant networks even though the attached systems are not critical to mission safety. This would relieve the valuable human resources onboard Space Station from time-critical, data systems maintenance chores.

The development of technologies for an integrated, fault tolerant network capable of supporting data, voice and video communications is the emphasis of a research program at Langley Research Center. An experimental network is currently under development to investigate relevant design issues. Of importance to the design of any network is the establishment of a set of rules which govern communications that occur on the net. This paper presents a description of the experimental network, describes a hierarchy of protocols for that network and addresses protocol issues for control and data communications on the net.

Experimental Network Description

One objective of the Langley data systems research and technology program is to develop a local area network which will serve as a testbed for various network experiments. Of particular interest is the evaluation of algorithms and techniques for high data rate communications and fault tolerant operation. Results derived from the experiments will provide a data base that will assist in defining future Space Station networks.

A critical design issue for the experimental network involved the selection of an appropriate topology. While numerous topologies are available[5] (bus, ring, star), most provide only one or possibly two routes between any message source and destination in the network. To obtain a sufficient degree of fault tolerance, it is essential that the network provide many alternate paths between attached systems. This allows information to be routed around or away from faulty elements. The mesh topology (Figure 1) possesses this attribute which is the primary reason it was selected for the network design.

The nodes within the experimental mesh network of Figure 1 will execute distributed control algorithms to manage the flow of data and control information in the net. They must support the attached hosts by establishing data paths through the network and by insuring accurate delivery of data to the hosts. In addition, the nodes will monitor the operation of neighboring nodes and of the connecting links to detect failures or errors in the operation of those elements. As a simple example, assume for the network of Figure 2 that the normal path of communication between nodes 3 and 6 has been broken. The two nodes detect this condition (neither receives a response from the other) and information is automatically rerouted through node 2. The testbed network will be used to investigate various algorithms for fault detection and recovery. These algorithms must provide the nodes with the ability to detect and isolate inoperative elements, babbling nodes or hosts and nodes or hosts which continually place erroneous information on the network.

Real-time, video communications onboard the Space Station will require a network whose data paths offer high throughput and minimum delay. This requirement for video cannot be obtained using packet switching concepts.[6] The delays associated with segmenting large files into packets and then reassembling them at the destination are excessive. One solution is to establish a dedicated, high-throughput circuit between communicating sites on the net. This technique is known as circuit switching. The experimental mesh network will be configured to support circuit switching (Figure 3) and the network nodes will be responsible for establishing a host to host circuit prior to the start of communications. That circuit will remain intact until all data transfers between the two points are complete. Ultimately, it is necessary to employ fiber optic links in the network to accommodate video data rates. In addition, research is underway to investigate the use of integrated optical switching arrays in the nodes. This would eliminate the delays associated with converting between optical and electrical signals at each intermediate node.

As illustrated in Figure 3, circuit switching will support voice and data communications in addition to the video. However, some forms of data communications, such as interactive or bursty data, are better suited to packet switching. Using a dedicated circuit for this type of communication results in less efficient utilization of the channel. The control information for this experimental network configuration will be carried in short, control packets. A control packet is issued by its source node and is stored and forwarded at each intermediate node until it reaches the intended destination. Control information falls into one of four general categories: Commands, inquiries, notifications and responses. Some examples include commands to configure links for circuit establishment, inquiries as to the status of other nodes or hosts, notifications of detected faults and responses to the above.

Hierarchy of Protocols

All network communications must be governed by a set of protocols. Adherence to these rules by communicating entities results in orderly data exchanges and allows dissimilar systems to communicate over the network in an understandable manner. In addition, protocols attempt to provide reliable data transfers over less than totally reliable mediums. While the network protocol discipline encompasses a wide range of design issues (e.g. physical connectors, routing, applications), several fundamental protocol functions are presented in Table I. The remaining sections of this paper address these critical functions for data and control communications on the experimental network.

The hierarchy of protocols for the experimental mesh network are presented in figure 4. The physical layer defines the physical, electrical, and functional characteristics of the network such as fiber optic links and connectors and the particular data encoding scheme. Layers 1 and 2 govern the flow of control packets around the network. When a packet arrives at a node, the destination address contained in the packet is examined. If the address matches that of the holding node, the packet is accepted, decoded and acted upon by the node. If the address does not match, the node refers to its local routing table and forwards the packet to a neighboring node in the direction of the destination. The transfer of packets between neighboring or adjacent nodes is controlled by the node-to-node protocol. The source to destination protocol layer defines the end-to-end procedure for control packet communications. The overall network control activity is transparent to the attached host devices.

After a node gains network access for its attached host, it issues control packets to nodes along the path to the intended destination host, ordering them to configure a physical circuit (figure 5). Once this point-to-point circuit has been constructed, data communications between hosts can begin as per the layer 3 protocol. This protocol controls both the host-to-host data exchange and communications occurring over the host/node interface. Host-to-host protocol procedures are, as much as possible, the responsibility of the source and destination support nodes.

The highest level of protocol is the user or process oriented layer. This defines, for example, how a user interacts with the various host systems. This paper focuses on layers 1 through 3 of the protocol hierarchy, since they involve the actual data and control communications on the network. No attempt has been made at this time to present a correspondence between these protocol layers and those of the 7 layer International Standards Organization's (ISO) reference model.[7] However, as experimental results are obtained and the network definition becomes more complete, and effort will be made to relate the hierarchy of protocols for the mesh net to the ISO model.

Node to Node Protocol

The node-to-node procedure manages the transmission of control packets between adjacent nodes. It provides a methodology for detecting packets which were corrupted over the connecting link, retransmitting packets which were in error, acknowledging error-free transmissions and detecting duplicate packets. Figure 6 presents flow diagrams for both the sending and receiving node. After transmitting a packet, the sender waits to receive an acknowledgment from the receiver. When the packet arrives at the receiver, a cyclic redundancy check (CRC) algorithm is applied to the packet to check for bit errors. If the transmission was error-free, an acknowledgment is returned to the sender and the sender discards its copy of the packet. If errors were detected, the receiver simply discards the packet. The transmitting node will eventually time-out of the wait mode and retransmit the packet. This positive acknowledgment scheme is similar to that of Arpanet.[8,9] It is important to note that the returned acknowledgment for a good transmission can also be corrupted. In this case, the sending node which is receiving the acknowledgment will discard it, time-out and retransmit what will be a duplicate copy of the original packet. A good node-to-node protocol must include a mechanism for detecting duplicate packets.

A method for duplicate detection is provided by the alternating bit protocol.[10] For this scheme, both the send and receive channels connected by a physical link maintain an odd/even (O/E) bit. The state of the transmitter O/E bit and that of the receiver are initially the same. The transmitter will include its O/E bit in the packet it sends to the receiver. Assuming the packet was correctly transmitted, the receiver examines the state of the senders O/E bit contained in the packet. If the O/E bit of the transmitter matches that of the receiver, the packet is accepted and the receiver complements its O/E bit. If the O/E bits do not match, the packet is discarded as a duplicate. Regardless, the receiver's O/E bit is returned to the transmitter as an acknowledgment. When the transmitter receives the acknowledgment, it compares the state of its own O/E bit to the state of the receiver's. If they do not match, the packet was successfully transmitted and acknowledged and the transmitter then complements its O/E bit to match that of the receiver. If the O/E bits match, the acknowledgment is ignored as a duplicate.

As an example, assume that the O/E bit of the transmitter, $[O/E]_X$, and that of the receiver, $[O/E]_R$, are both initially '0'. The transmitter successfully sends a packet containing $[O/E]_X$ to the receiver. Since both O/E bits match, the receiver accepts the packet, complements $[O/E]_R$ to a '1' and returns $[O/E]_R$ as an acknowledgment. Now assume that this packet containing the acknowledgment is corrupted on the link. The transmitter discards it and eventually retransmits a duplicate packet. Upon receipt of this packet, the receiver detects a mismatch between $[O/E]_X$ and $[O/E]_R$ and therefore ignores the duplicate packet. The receiver will once again return $[O/E]_R$ as an acknowledgment.

This time the acknowledgment gets through to the transmitter and it recognizes that $[O/E]_X$ is a '0' and $[O/E]_R$ is equal to '1'. Therefore, the transmitter accepts the acknowledgment and sets $[O/E]_X$ equal to '1' in preparation for the next transmission.

To obtain greater link efficiency and utilization, more than one logical channel must be assigned to the physical link connecting two nodes (Figure 7). This allows consecutive packets to be forwarded without waiting to receive an acknowledgment for an earlier packet. In addition, acknowledgments can be "piggybacked" onto control packets headed in the opposite direction of the packets being acknowledged. If none is available, short, dedicated packets will be used to carry the acknowledgements back to the transmitter.

It is desirable to establish an expression from which the delay associated with forwarding a packet from node-to-node can be computed. Let T_X be the time between the arrival of a packet at one node to the arrival of that packet at the next adjacent node. Then, assuming no retransmission:

$$T_X = T_P + T_Q + T_C + T_D \qquad (1)$$

where T_P = node processing time

T_Q = packet wait time

T_C = data clocking time

T_D = channel propagation delay

The node processing time is the time required to perform such functions as error checking and route selection. T_Q is the delay the packet experiences waiting in the queue for an available output channel. T_C is the time required to clock the data out on the link and is defined as the length of the packet in bits divided by the transmission rate (bits/sec). The end-to-end packet delay can be expressed as:

$$\sum_{i=1}^{N} T_{X(i)}$$

where N is the number of intermediate nodes between the source and destination nodes.

To transmit a control packet from one node to the next, a mechanism must be established that provides synchronization, bit error detection and data transparency. Bit oriented High-Level Data Link Control (HDLC) addresses these functions. The frame structure for bit oriented HDLC is presented in Figure 8. The flag is a unique 8-bit sequence that delimits the beginning and end of the packet. The start flag provides the synchronization necessary to locate the packet header and the ending flag defines the location of the frame check sequence. The frame check sequence is either a 16 or 32 bit CRC code which is employed by the receiving node to detect packets that were corrupted on the link by bit errors. Data transparency is realized by applying a zero bit insertion/deletion algorithm to bits between the two flags. This algorithm has the transmitter insert a '0' after it encounters five consecutive ones in the packet. Therefore, a flag will not occur in the packet header, control data or frame check sequence. The receiver then removes any '0' it receives after five consecutive '1' bits.

A critical issue for the node-to-node design involves the size of the store and forward buffers in each node. The node storage must be of sufficient capacity to assure every arriving packet of a temporary buffer space. This is necessary to avoid flow control problems between nodes such as the case where two adjacent nodes are unable to forward packets to each other, due to lack of buffer space, and are forced to lose all incoming packets (store and forward lockup).[8] Node buffers must be sized for peak traffic situations with additional buffers left over to guarantee input and output to every arriving packet.

Source to Destination Node Protocol

The interaction between the source node and destination node for the end-to-end transmission of control packets in the experimental mesh network is defined by the source to destination protocol. The responsibilities of this level of protocol are listed in Table II. As in the node-to-node process, packets which were corrupted by bit errors must be detected at the receiver or destination node and these packets must be retransmitted by the source. A positive acknowledgment scheme will again be employed to notify the source node of a successful transmission. In addition to the corruption of packets, there is also the possibility that a packet could get lost on its journey to the destination. This can occur when an intermediate node goes down after receiving and acknowledging a packet but before forwarding it to the next node. A packet could also be "boxed in" by existing host to host circuits and be unable to reach its destination. Regardless of the cause, the source node would fail to receive an acknowledgment for the lost packet. It would eventually time out and retransmit a copy of the packet to the destination. As a means for detecting duplicate packets, the end-to-end procedure requires that the source node assign a sequence number to each packet prior to transmission. This number uniquely identifies each packet so that if one arrives from a prior point in the sequence it is recognized as a duplicate. For example, when the destination node receives a packet, it compares the sequence number (SN) to its expected sequence number (ESN) for the particular source of the transmission. If the two are equal, the destination node accepts the packet. If SN < ESN, the packet is discarded as a duplicate. It is essential that each source-destination pair in the net maintain synchronization between SN and ESN.

The source to destination protocol is illustrated by the flow diagrams of Figure 9. The source node assigns the appropriate sequence number to a packet prior to transmission. After the packet is sent, the transmitter starts its timer and waits for an acknowledgment. If the packet is received error-free at the destination node, the node examines the sequence number of the packet. If the packet is accepted (SN = ESN), the expected sequence number of the destination is incremented by one. The expected sequence number is then returned as an acknowledgment to the source. This will acknowledge not only the last packet transmitted but all packets sent from the source to that destination of $SN \leq ESN - 1$. The return of ESN as an acknowledgment provides periodic resynchronization between SN and ESN for the particular source-destination pair. If the packet carrying the acknowledgment is corrupted on its way to the source, the source will discard it, time out and retransmit a duplicate packet. The duplicate will have an SN < ESN and will therefore be ignored by the destination. The destination will once again send ESN back to the source as an acknowledgment. A corrupted control packet will also be detected and discarded at the destination, but no acknowledgment is returned. Corrupted packets are recovered via the time out, retransmission mechanism of the source node.

The sequence with which control packets arrive and are operated on by the destination node can be essential to the proper operation of the network. The protocol illustrated in Figure 9 maintains this sequence.[11] The sequence of packet arrival can be further assured by defining a window size of one for the network, i.e. only one packet can be outstanding without acknowledgment between any source-destination pair. For example, if node A sends a packet to node C, A must refrain from sending another packet to C until C acknowledges the previous one. As the window size is increased, so is the likelihood of packets arriving out of sequence. This is caused by packets taking different routes to the same destination or packets being discarded due to bit errors. The sequential protocol of Figure 9 specifies that only packets which arrive in sequence (SN = ESN) are accepted, all others are discarded. This is suitable for use with narrow window specifications; but, as the window size expands, an inordinate number of source retransmissions may be required (all packets of SN > ESN are thrown away). Considering the hand shaking nature of control packet communications on the experimental network (e.g. inquiry, response, respond to response), a window size of one should not be restrictive for the source/destination exchange.

The final task listed in Table II for the source to destination protocol involves establishing provisions for flow control at the destination node. If a node in the network continues to receive control packets faster than it can decode and act upon them, its internal buffers will eventually fill. The node is then unable to accept additional packets. One course of action for the destination node is to simply discard a packet if no buffer is available and rely on the time out, retransmission mechanism of the protocol to recover the packet. This is a tolerable solution if there is a high probability that the packet generated by the second transmission will find a free buffer at its destination. As an alternate method, the destination could place the originator of the discarded packet on a reservation list and send to the source an "allocate" message when a buffer becomes available.[8] The source would retransmit the packet immediately after receiving the allocation. While this approach introduces additional complexity, it potentially reduces the packet recovery time since the source node can retransmit prior to timing out.

Since the packet traffic on the experimental mesh network is limited to control functions (at least in the initial design), node congestion is likely to be infrequent and short lived. In addition, the control information will be composed of brief, single packet messages to, as much as possible, avoid taxing the storage capacity of the nodal buffers.

A possible format for the control packets is presented in Figure 10. The packet is composed of a header, the text or network control information and an end-to-end CRC code. The frame check sequence described in the node to node procedure only tests for errors that occur on the connecting link. Therefore, it is necessary to apply a second CRC code to the packet at its destination to detect errors inflicted during the end-to-end journey. This test takes into account the corruption of packets by the nodes themselves. The header contains several packet control fields. The first field holds the odd/even bit of the transmitter as defined by the alternating bit protocol. This is followed by the number of the logical output channel and the node-to-node acknowledgment bits (one per channel). The "receive ready" (RR) is a one bit field that identifies the packet to be a dedicated acknowledgment for a source to destination transmission. The RR packet is used to carry the expected sequence number back to a source when no control packet is available for that task at the destination node. No text is included in the RR packet. The next two fields specified by the format are the sequence number (SN) of the packet and the expected sequence number (ESN) of the transmitting node. They are followed by an internal timer which keeps track of how long the packet has been traveling in the network. This allows a node to detect and kill old packets which have been wandering excessively around the net without reaching their destination. The intention is to avoid the possible confusion created by a late arriving packet which was originally assumed to be lost. The trace field is used by the intermediate nodes to determine which nodes have received the packet. The packet will not be returned to any node it has already visited. This prevents looping or ping-ponging of packets between nodes. The last two fields of the packet are reserved for the addresses of the destination node and the source node respectively.

Host to Host Protocol

The host to host protocol for the experimental mesh network establishes procedures which govern the communication between hosts over a dedicated circuit (figure 5). This protocol layer insures error free delivery of data to the destination host, supports large file transfers (data and video) as well as interactive data and voice communications and provides flow control. The hosts rely on their supporting nodes to attend to these procedures so that the process is, as much as possible, transparent to the host systems. The protocol also addresses the exchange of data over the host-support node inferface.

A versatile menu of procedures for point-to-point communications are offered by the International Organization for Standardization High Level Data Link Control (HDLC) protocol.[11] These procedures can be tailored to meet the requirements for host communications on the experimental network. HDLC addresses both unbalanced configurations, where stations act in a master-slave relationship, and balanced configurations, where stations have equal status. The balanced configuration will be assumed for all host to host exchanges on the network.

The format for an HDLC frame is presented in figure 11. The flags, frame check sequence (FCS) and data transparency algorithm are identical to that described for bit oriented HDLC (figure 8). The data to be transmitted from host to host is contained in the information field. The frame level control field consists of two 8-bit bytes (non-extended mode) one of which is an address and the other conveys control information between host support nodes. The address can be that of the destination or originator of the frame depending on whether the frame is a command or response, respectively. This addressing scheme allows the destination to distinguish between commands and responses since some frames can be either. There are three possible classes of HDLC frames and the control field specifies to which class the frame belongs. The information frame carries the data across the circuit to the destination. Supervisory frames control the data flow and support error recovery. Unnumbered frames are used primarily for initialization and termination of a communications link as well as for status reporting.

The control field for each HDLC frame class is presented in figure 12. If the first bit of the field is an '0' the frame belongs to the information class. If it is a '1' the frame is either supervisory or unnumbered depending upon the state of the second bit. The three bit N(S) code specifies the sequence number of the information frame (0 through 7) and N(R) is the expected sequence number for data flow in the opposite direction. As in the source to destination node procedure, the expected sequence number is used by HDLC to acknowledge error-free frames. The poll/final bit (P/F) controls master-slave communications between secondary and primary stations in the unbalanced configuration. This bit is also employed in one of the HDLC error recovery schemes. The S field contained in the supervisory frame identifies one of the 4 types of supervisory frames. The 5-bit M field defines one of 32 types of unnumbered frames; however, about 20 frames have actually been defined.

A list of HDLC frames and their mnemonics is presented in Table III. The "receive ready" (RR) acknowledges previously received I frames and indicates that the particular destination is able to receive additional frames. The "receive not ready" (RNR) acknowledges I frames but signals the sender that additional frames cannot be received at that specific time. This supervisory frame is used in HDLC for flow control. "Reject" (REJ) and "selective reject" (SREJ) both report errors in received I frames and request retransmissions from the source. HDLC error recovery mechanisms will be described later in this paper. There are several "set mode" commands in the unnumbered class which initialize stations for normal operation (primary-secondary stations), for asynchronous operation and for balanced configurations. The "set initialization mode" (SIM) command initializes station specified procedures and its details are defined by the application. The "disconnect" (DISC) command is employed to terminate communicatons over the link. For a more detailed description of these and other HDLC frames, the author suggests references 11, 12 and 13.

An example of a possible host to host data transfer on the experimental network using the HDLC protocol is illustrated in figure 13. After a dedicated circuit has been established between two host support nodes, S_1 and S_2, S_1 issues a SIM command. This initialization command sets sequence numbers and expected sequence numbers [N(S) and N(R), respectively] at both locations to zero and might also involve reserving storage at the destination host or matching speeds of the transmitter and receiver. The destination support node must return an "unnumbered acknowledgement" (UA) for the SIM command to cover the possibility of the command being corrupted on the link and discarded by S_2. If S_1 fails to receive an UA response before timing out, it retransmits the SIM command. After initialization is complete, the source support node, in the example, transmits the data from its host to S_2 enclosed in seven information frames (I frames). It so happens that the HDLC protocol employs a window size of seven which means that the source will cease transmitting after 7 frames and will wait until it receives an acknowledgment before sending additional I frames. Assuming that all I frames arrive at S_2 error-free, the destination support node passes the data to its host and returns to S_1 a "receive ready" (RR) frame with N(R) equal to 7. This will acknowledge all transmitted information frames, I_0 through I_6. Since its host has no additional data to send, the source support node issues a "disconnect" (DISC) command to S_2. The communication is terminated when S_2 acknowledges the DISC command. At this time, the network nodes can disconnect the host to host circuit.

Figure 14 illustrates the use of the "receive not ready" frame for HDLC flow control. Support node S_1 transmits 6 I frames from its attached host to support node S_2. S_2 accepts I_0 through I_3 but is unable to accept I_4 and I_5. This might be due to the fact that S_2 is receiving frames faster than it can transfer error-free frames to is host. Support node S_2 sends a "receive not ready" to S_1 which acknowledges the I frames S_2 was able to accept (up to I_3). After waiting for a specified period of time, S_1 resends I_4. This periodic retransmission is necessary to account for the situation where a "receive ready" (RR) may have been sent by the destination but was corrupted on the link. Since S_2 is still unable to accept additional I frames, it responds to I_4 with another RNR frame. When the congestion has cleared at the destination, support node S_2 sends a RR frame to S_1 and S_1 resumes normal transmission of information frames.

HDLC provides four techniques for the recovery of I frames which were corrupted by bit errors and discarded at the destination. While it is doubtful that any system would employ all of the available error recovery schemes some subset of the four techniques would be selected depending upon the application requirements. The technique which would most certainly be employed by all applications is the time out mechanism. A transmitting station starts its time out counter as soon as it transmits the first I frame. The receipt of an acknowledgment for some of the I frames restarts the counter and the counter stops when all frames are acknowledged. The counter will then restart upon transmission of a new I frame. Should the counter time out, the source retransmits all unacknowledged I frames.

The second technique employs the supervisory frame "reject" (REJ) sometimes called the unselective reject. When a destination receives an out-of-sequence frame (expects X, gets X+1) it realizes that the expected I frame was corrupted on the link and lost. The destination returns a "reject" frame containing its expected sequence number [N(R)] to the source of the I frames. It will then discard all I frames until it gets the expected frame. When the source station gets the REJ, it accepts the acknowledgment for all I frames up to N(R) - 1 and retransmits all information frames from N(R).

The frame reject procedure can be made more efficient in terms of minimizing retransmissions by using the selective reject technique. In this approach, a destination that expects frame X but gets X+1 accepts X+1 and all subsequent I frames. It then issues a "selective reject" (SREJ) frame to the source with N(R) equal to X. After receiving the SREJ frame, the source retransmits only the I frame of sequence number X.

The final HDLC error recovery technique is poll/final bit check pointing. This procedure allows a source station to inquire as to which of its I frames have been successfully received. The sending station sets the poll bit in a command (e.g. an information frame) and transmits the command to the destination. The receiving station must reply as soon as possible using an I or supervisory frame with the final bit set. This response contains the expected sequence number, N(R), which will acknowledge all correctly received frames from the sender. The source station examines the response and if all of its transmitted frames are not acknowledged, the source begins retransmission from I frame number N(R).

Figure 15 provides an example of HDLC and its error recovery mechanisms employed for a large data file transfer. This illustrates a possible host to host exchange on the experimental mesh network. To achieve the maximum throughput for the data transfer, the sending station (ST1) must continually transmit consecutive I frames without interruption. If the receiving station (ST2) fails to return an acknowledgment for at least some of the I frames before the HDLC window size is reached, ST1 will cease transmitting and wait for the acknowledgement. In figure 15a, acknowledgments arrive at the sending station before the transmission of every 7th I frame is complete; therefore, there are no "gaps" in the transmitted data stream. Of course, this example assumes that the receiver has no difficulty accommodating the rate of arrival of the I frames. In addition, all information and supervisory frames are considered to be error-free.

Figure 15b illustrates the recovery mechanism for the situation where an I frame is corrupted on the communications channel. Station 2 detects a bit error in I_1 and immediately discards that information frame. When I_2 arrives at ST2, it is recognized as an out-of-sequence frame (ST2 expected I_1 but received I_2). The receiving station accepts subsequent frames but sends a "selective reject" to ST1 which identifies the missing frame. The sending station retransmits I_1 as soon as possible after receiving the SREJ frame. Supervisory frames can also be corrupted by bit errors. In figure 15c, ST2 returns a "receive ready" acknowledgment to ST1 which becomes corrupted on the connecting link. ST1 will discard the RR frame, cease transmitting since the HDLC window size has been reached and eventually time out waiting for an acknowledgment. After timing out, ST1 resends the last transmitted I frame, I_6, with the poll bit set to a '1'. Upon receipt of this poll command, the receiving station sends another "receive ready" frame, with the final bit set, which acknowledges all I frames transmitted by ST1. The sending station can then resume normal data transmission.

The high data rate imposed by real-time, video data transfers are likely to prohibit the use of HDLC error recovery schemes for that class of network communications. The overhead associated with frame retransmissions is excessive for this application. In fact, the mere acquisition of continuous data frames by the receiving node at 100 megabits per second and their transfer to the attached host represents a significant technological challenge. Fortunately, it is not necessary for large video files to be totally error free to convey significant information when viewed on a display. Therefore, a node receiving video data on the network might simply keep track of the number of corrupted frames in the transmission and take corrective action (e.g. path reconfiguration) if the number of errors exceeds some predetermined threshold.

Examples of an interactive, full duplex data exchange between network hosts are presented in figure 16. The information frames carry acknowledgments (N(R)) for I frames traveling in the opposite direction and the selective reject mechanism is once again used to recover corrupted I frames. In figure 16a, the second I frame sent from station 1 to station 2 (I_1) is corrupted on the link. This is detected by ST2 after it receives the out of sequence frame I_2. ST2 returns a SREJ with N(R) equal to 1 to request a retransmission of I_1 from ST1. The I frame transmitted by ST2 after it receives the retransmission (I_4) will contain an acknowledgment for all outstanding I frames from ST1.

In figure 16b, the SREJ sent by ST2 to recover I_1 is also corrupted. Station 1 will cease transmitting after it has sent the maximum number of unacknowledged frames (HDLC Window = 7). After timing out, ST1 resends the last transmitted I frame, I_7, with the poll bit set. ST2 responds with a retransmission of the SREJ which prompts station 1 to resend I_1. After receiving I_1, ST2 can provide an up-to-date acknowledgment to ST1 in a subsequent I frame (I_2'). The two stations can then resume their normal prime data exchange.

The host to host protocol must also address the transfer of data between the host and its support node. The host/node interface for the experimental network will consist of short, parallel lines for high throughput data transfers. Additional lines will be required to convey control information. A typical control exchange between the host and node for a data word transfer might be, "Prepare to Receive a Word"/"Word Received Correctly"/"Good." The procedure for host/node communications must also define a technique for detecting bit errors in the delivered word and for resending the word when errors are discovered. A simple error detection algorithm (possibly a parity check) may be sufficient since the probability of bit errors is reduced by the short length of the data path and the reduced data rate per line achieved via parallelism. Finally, a flow control method is required to allow the host or node to halt the data transfers until it can "catch up" with the sender. The HDLC "receive not ready" response can be employed for this purpose.

Concluding Remarks

A hierarchy of protocols has been devised for an experimental mesh network. This local area net will serve as a testbed for various network experiments such as the evaluation of algorithms for fault tolerant operation and integrated communications. Control information on the network will be carried in dedicated, control packets. These packets will be received and forwarded toward their destination by each intermediate node between the source-destination node pair. All data, voice and video communications will be supported by circuit switching, i.e. a physical circuit will be constructed by the network nodes between two communicating hosts prior to the start of a host to host data exchange. The rationale for this design decision is based upon the high bandwidth requirement of the real-time video transmissions. However, interactive or "bursty" data communications are better suited to packet switching since it provides improved channel utilization for that type of data. Therefore, techniques are currently under investigation which would incorporate packet switching into the experimental network for interactive data exchanges. Many of the procedures described in this paper for control packets could also apply to the delivery of data packets.

The rules which govern the flow of control and data through the network are set forth in layers 1, 2 and 3 of the five layered hierarchy of protocols. In defining these protocols, particular emphasis was placed on robust algorithms for error control and reliability. Ultimately, these procedures must be implemented within the mesh network to determine their impact on network performance in terms of message delay and data throughput.

References

1. Hendricks, H. D.; Murray, N. D.: "Wavelength Division Multiplexing for Future Space Station Data Systems." Proc. SPIE, Vol. 434, San Diego, CA, August 1983.

2. Hendricks, H. D.; Murray, N. D.: "Wavelength Division Multiplexing Fiber Optic Data System Utilizing Both AlGaAs and InGaAsP Semiconductor Laser Wavelengths for Future Space Station Applications." CLEO Technical Digest, Anaheim, CA, June 1984.

3. Thurber, K. J.; Freeman, H. A.: Tutorial: Local Computer Networks, Computer Society Press, Los Alamitos, CA, 1980.

4. Proceedings of the Local Area Communications Network Symposium, Meisner, N. B. and Rosenthal, R., Editors, Cosponsored by the Mitre Corporation and the National Bureau of Standards, Washington, DC, May 1979.

5. Davies, D. W.; Barber, D. L. A.; Price, W. L.; Solomonides, C. M.: Computer Networks and Their Protocols, Ch. 3, John Wiley and Sons Ltd., 1979.

6. Davis, D. W.; et. al.: Computer Networks and Their Protocols, Ch. 2, John Wiley and Sons Ltd., 1979.

7. "Reference Model of Open Systems Interconnection," International Standards Organization Documents ISO/TC97/SC16/N227 and ISO/TC97/SC16/N309.

8. McQuillan, J. M.; Walden, D. C.: "The Arpa Network Design Decisions." Computer Networks, Vol. 1, No. 5, Sept. 1977.

9. Kimbleton, S. R.; Schneider, G. M.: "Computer Communication Networks: Approaches, Objectives and Performance Considerations." Computing Surveys, Vol. 7, No. 3, Sept. 1975.

10. Bartlett, K. A.; Scantlebury, R. A.; Wilkinson, P. T.: "A Note on Reliable Full-Duplex Transmission Over Half-Duplex Links." CACM, Vol. 12, No. 5, 1969.

11. Davis, D. W.; et. al.: Computer Networks and Their Protocols, Ch. 6, John Wiley and Sons Ltd., 1979.

12. "Data Communications - High Level Data Link Control - Elements of Procedures" ISO 4335, International Standards Organization, 1979.

13. Brodd, W. D.: "HDLC, ADCCP and SDLC: What's the Difference?" Data Communications, August 1983.

TABLE I

Protocol Functions

● Synchronization	- Coordinate Sender and Receiver Prior to Data Transfer
● Delimitation	- Denote Start and End of Message
● Data Transparency	- Permit Any Bit Sequence to be Included in Data
● Data Transfer	- Support Controlled Data Transfer from Sender to Receiver
● Error Control	- Insure Accurate, Reliable Data Delivery
● Flow Control	- Compensate for Excessive Arrival Rate of Data at Destination

TABLE II

Responsibilities of the Source to Destination Node Protocol

- Detect Bit Errors in the Delivered Packet
- Retransmit Packets Which Were Corrupted on Their Journey to the Destination
- Return Acknowledgments to Source Node for Good Transmissions
- Recover Lost Packets
- Detect Duplicate Packets
- Provide Flow Control at the Destination Node

TABLE III

HDLC Frames

Class	Name	Mnemonic
Information		I
Supervisory		
	Receive Ready	RR
	Receive Not Ready	RNR
	Reject	REJ
	Selective Reject	SREJ
Unnumbered		
	Set Normal Response Mode	SNRM
	Set Asynchronous Response Mode	SARM
	Set Asynchronous Balance Mode	SABM
	Set Initialization Mode	SIM
	Request Initialization Mode	RIM
	Disconnect	DISC
	Unnumbered Poll	UP
	Reset	RSET
	Unnumbered Information	UI
	Exchange Identification	XID
	Unnumbered Acknowledgment	UA
	Disconnect Mode	DM
	Request Disconnect	RD
	Frame Reject	FRMR
	Test	TEST

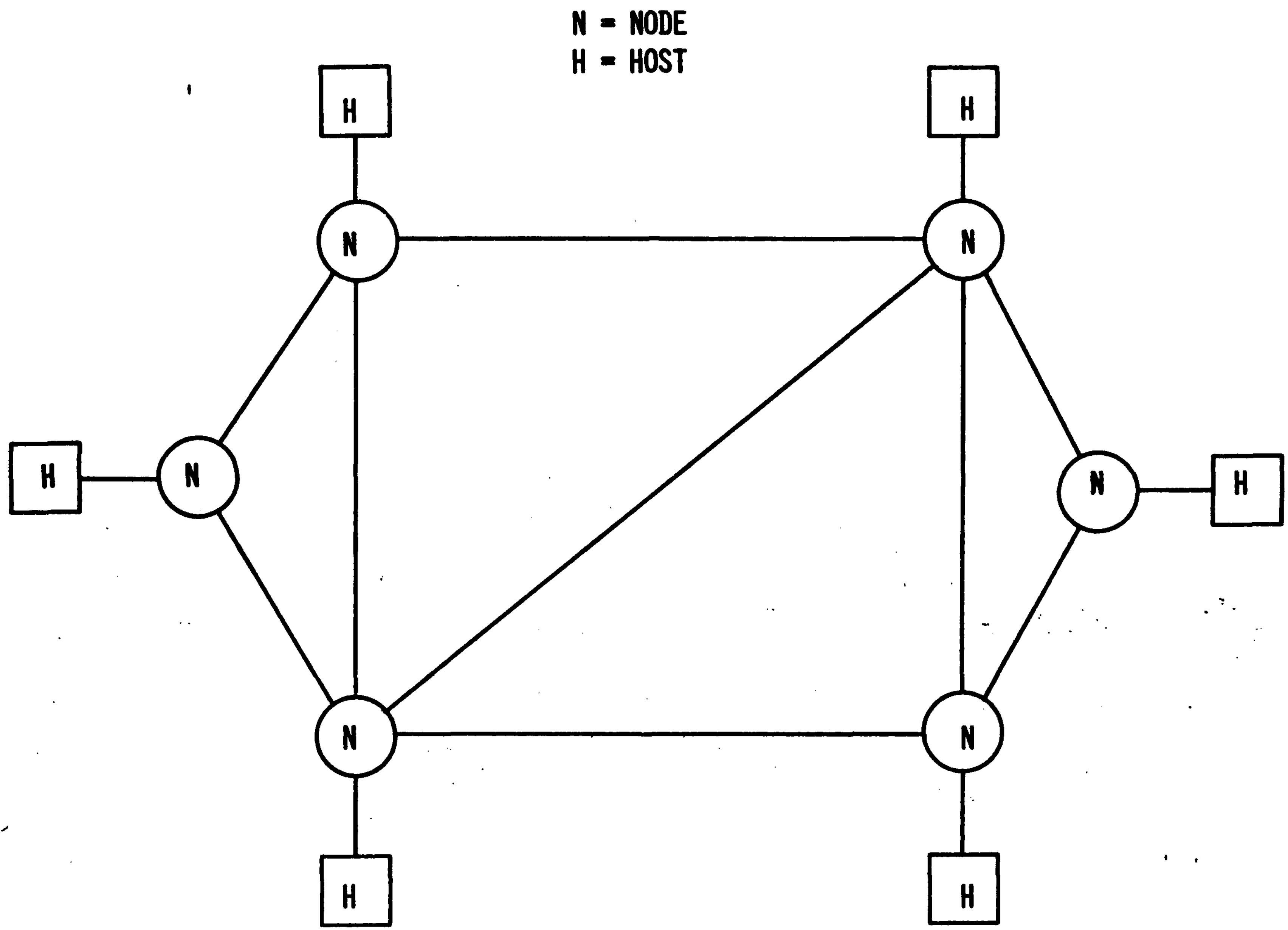


FIGURE 1. THE MESH TOPOLOGY PROVIDES ALTERNATE PATHS FOR FAULT TOLERANCE

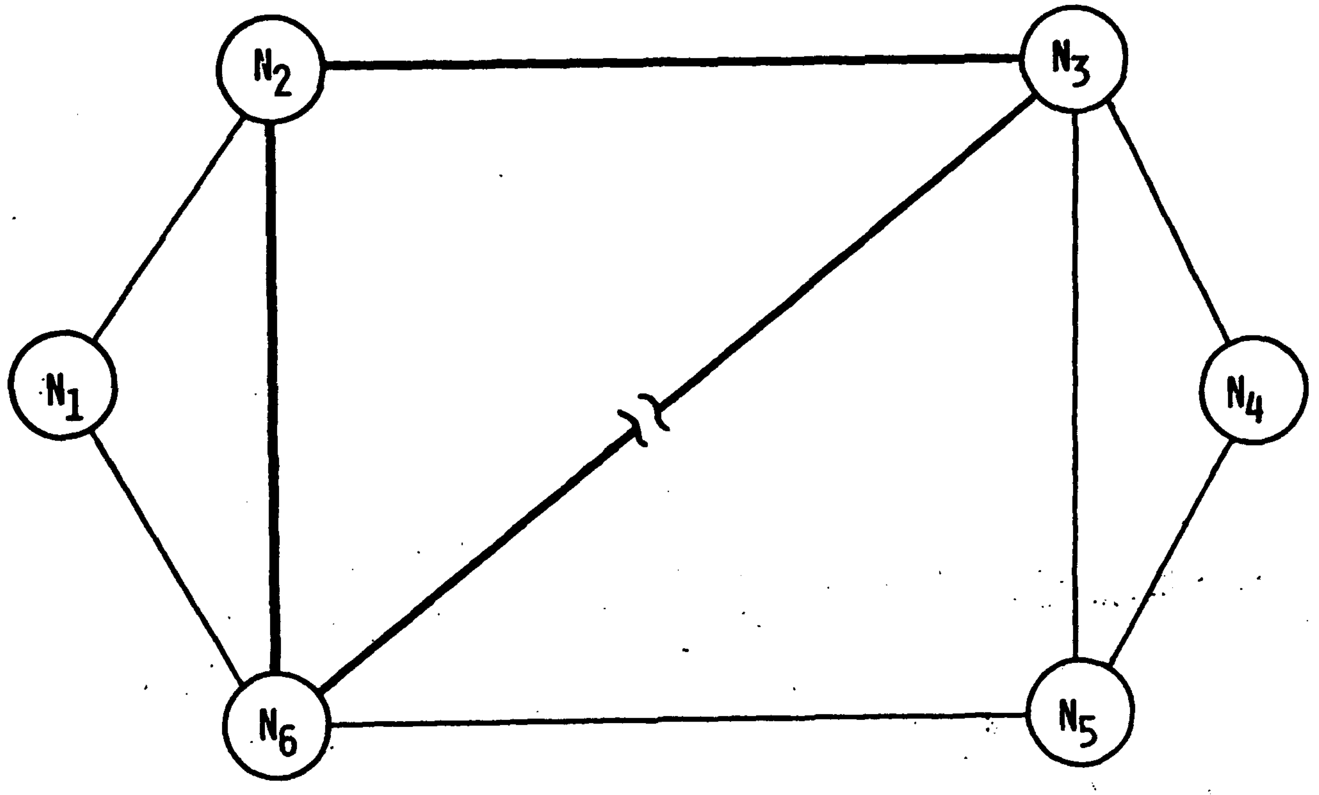


FIGURE 2. NODES 3 AND 6 DETECT FAULTY LINK AND REROUTE COMMUNICATIONS THROUGH NODE 2.

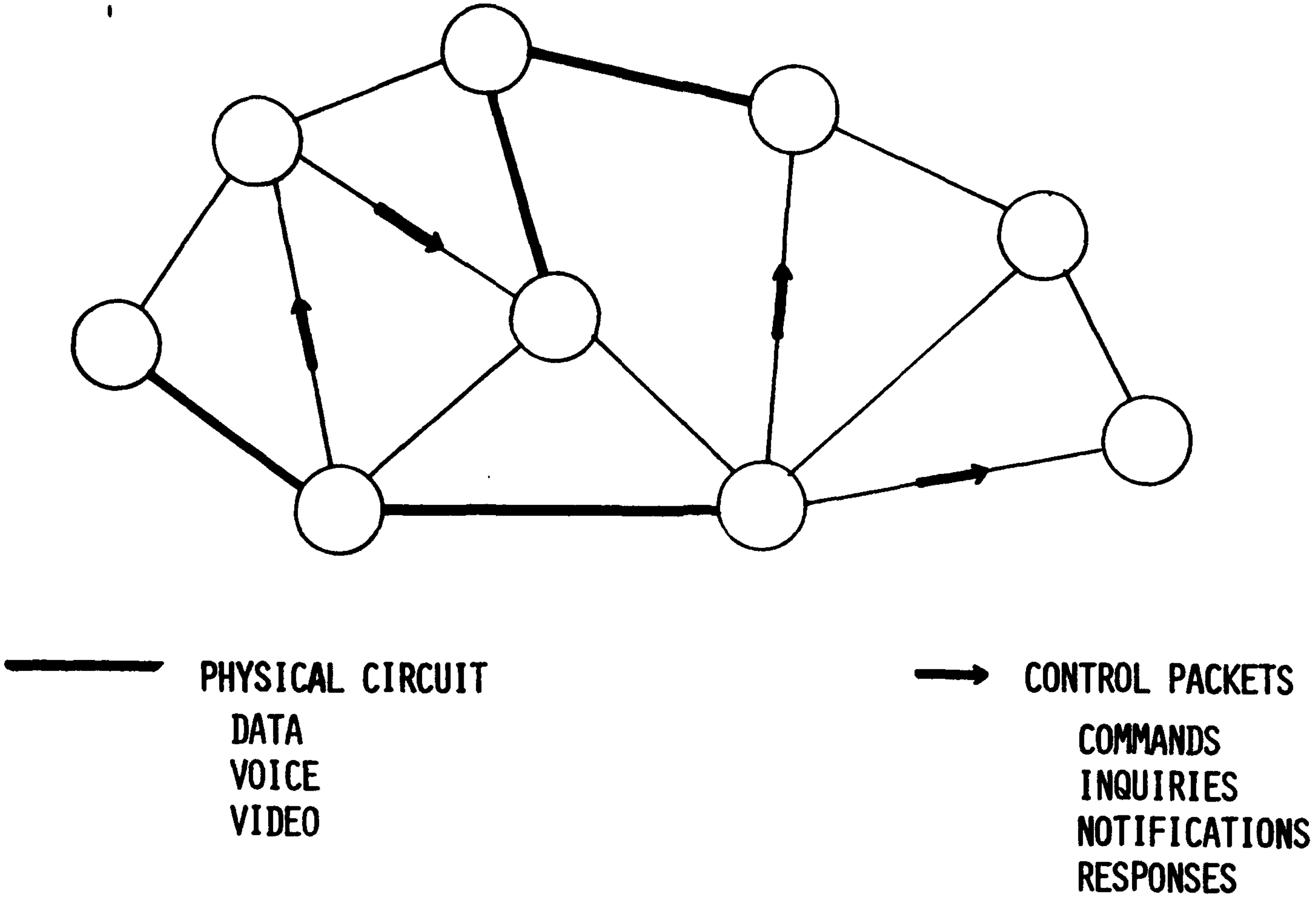


FIGURE 3. EXPERIMENTAL NETWORK CONFIGURATION EMPLOYING CIRCUIT SWITCHING FOR HIGH THROUGHPUT COMMUNICATIONS

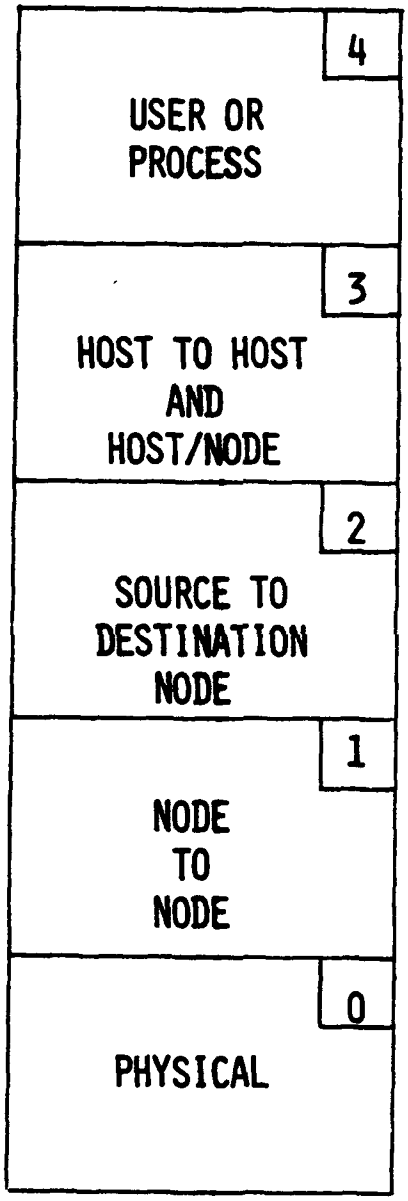


FIGURE 4. HIERARCHY OF PROTOCOLS FOR THE EXPERIMENTAL MESH NETWORK

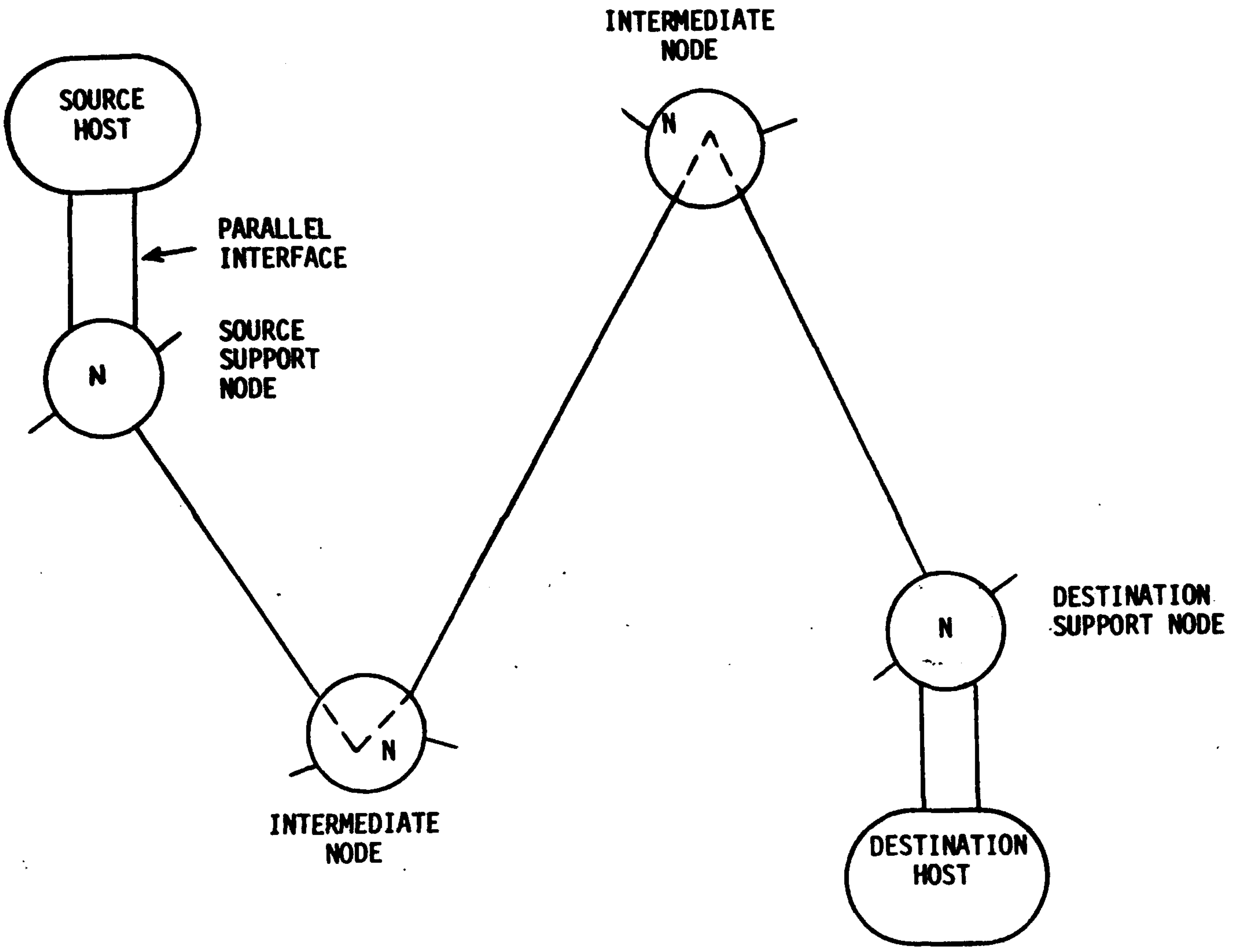
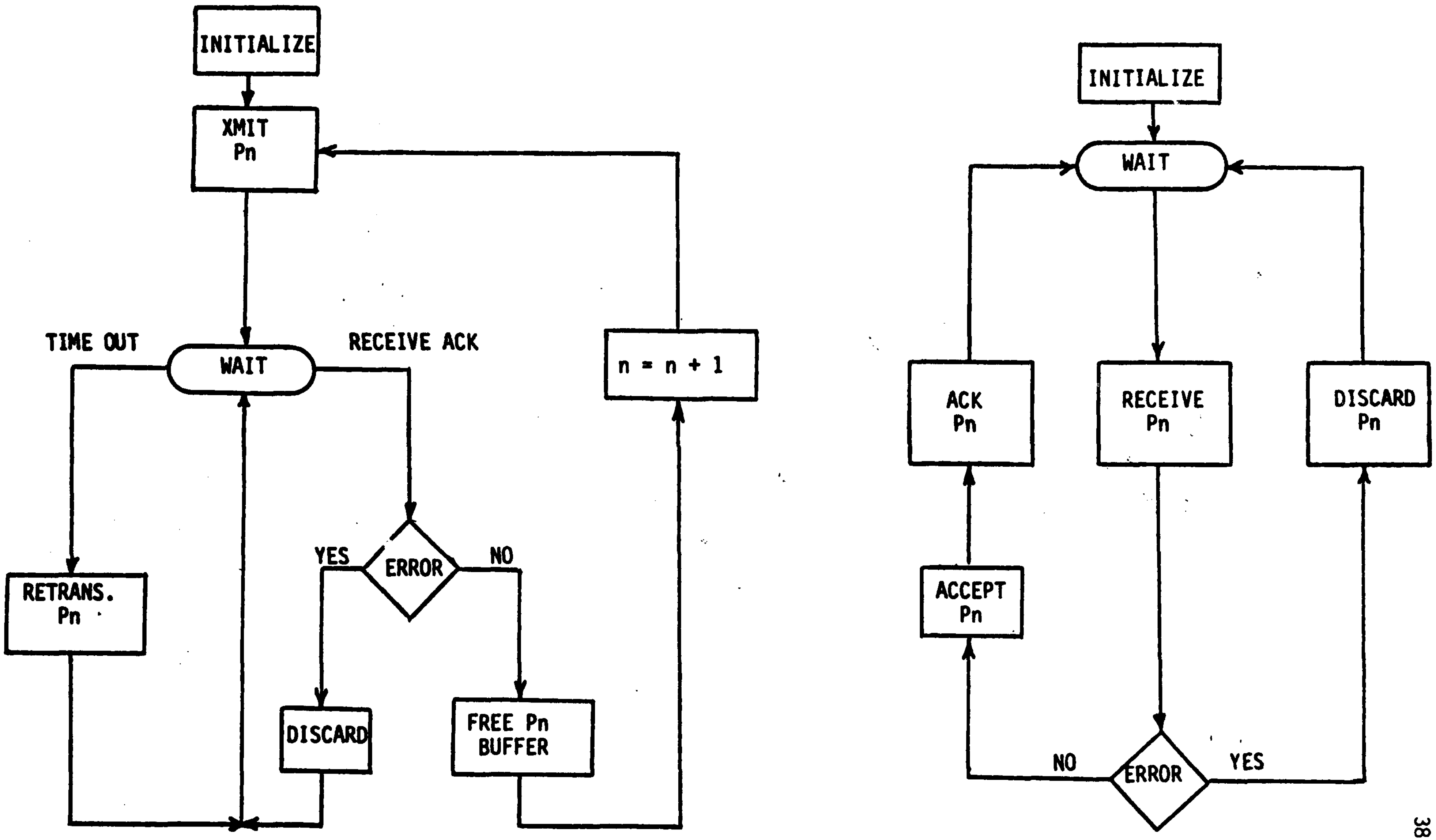


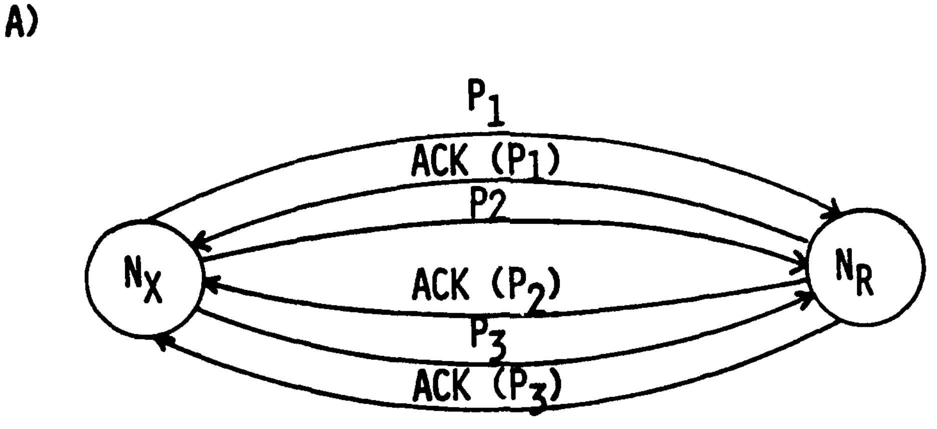
FIGURE 5. CIRCUIT SWITCHED PATH FOR HOST TO HOST COMMUNICATIONS



(a) SENDING

(b) RECEIVING

FIGURE 6. NODE TO NODE PROTOCOL FLOW DIAGRAMS



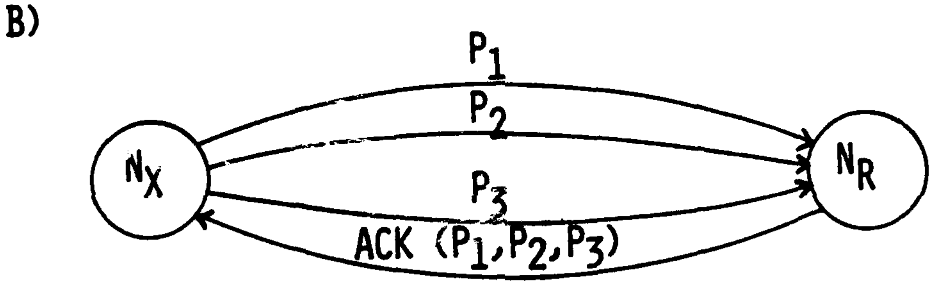
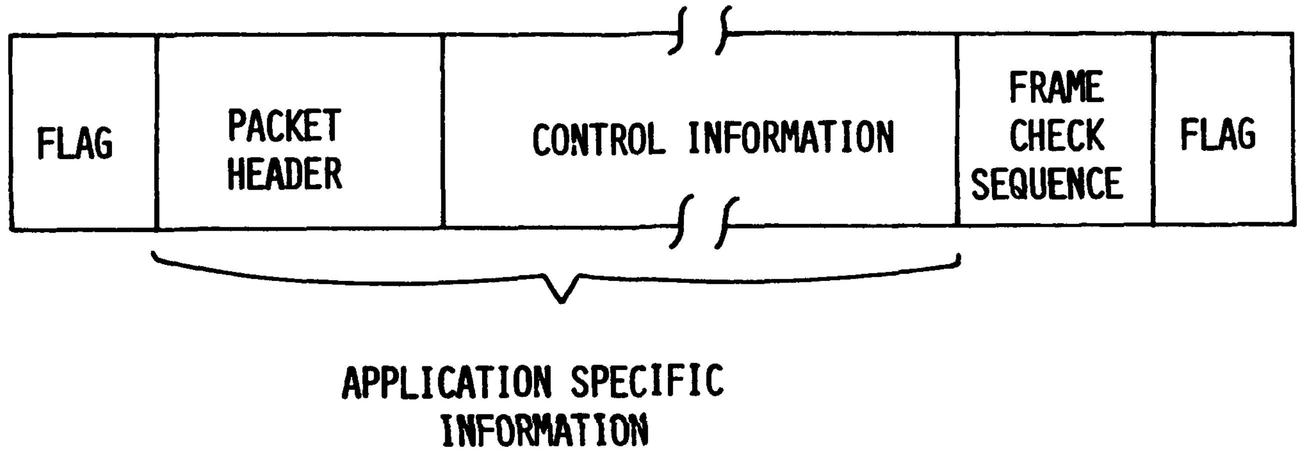
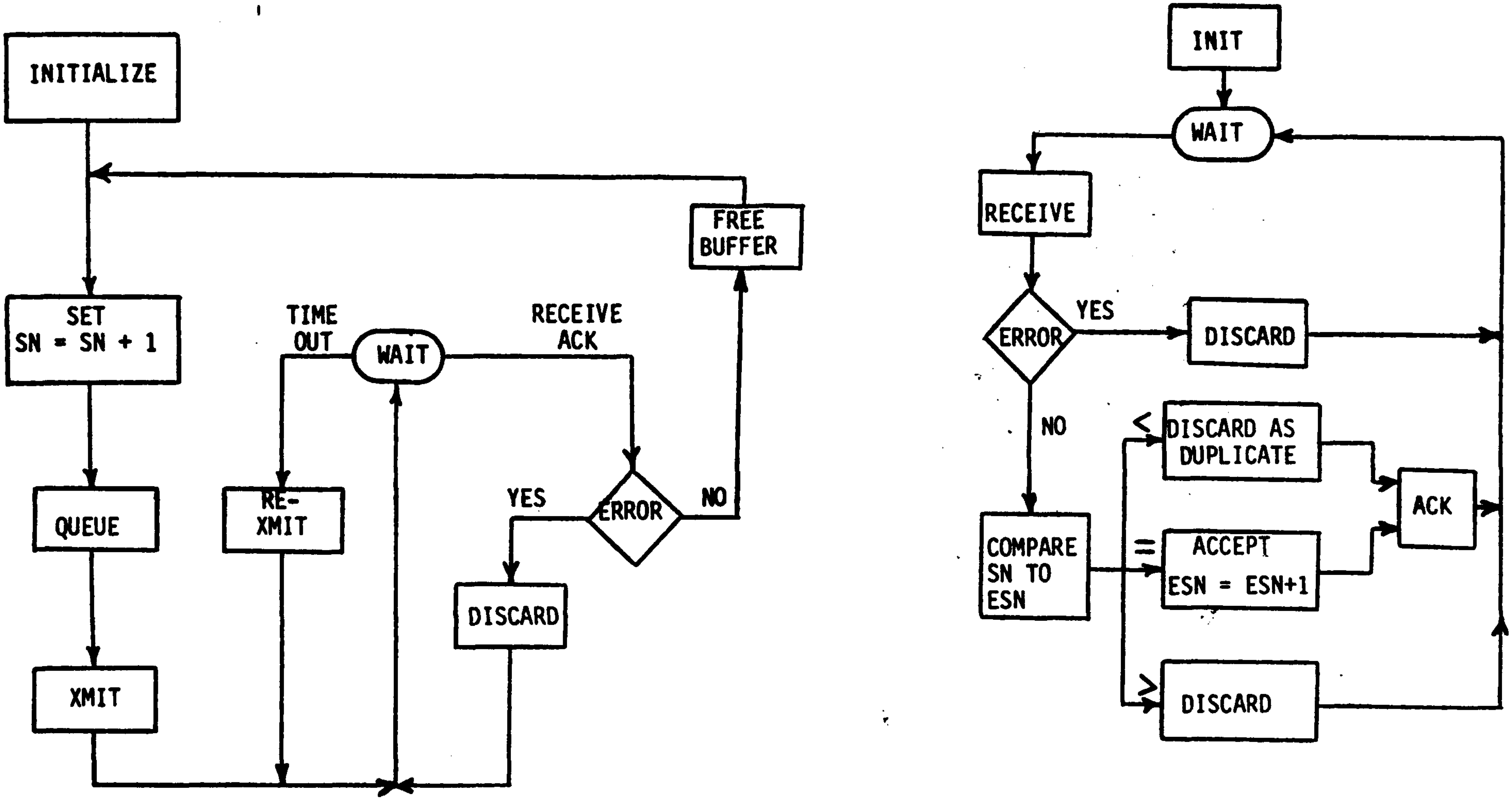


FIGURE 7. NODE TO NODE EXCHANGE. A) ONE LOGICAL CHANNEL. N_X MUST RECEIVE AN ACKNOWLEDGMENT FOR EACH PACKET BEFORE SENDING THE NEXT PACKET. B) THREE LOGICAL CHANNELS. N_X CAN SEND THREE CONSECUTIVE PACKETS BEFORE RECEIVING ACKNOWLEDGMENTS.



FLAG = '01111110'; DELIMITERS AND SYNCHRONIZATION
FRAME CHECK SEQ. - 16 OR 32 BIT CRC CODE
DATA TRANSPARENCY - ZERO BIT INSERTION/DELETION ALGORITHM

FIGURE 8. FRAME STRUCTURE - BIT ORIENTED HDLC



a) SENDING

b) RECEIVING

FIGURE 9. SOURCE TO DESTINATION PROTOCOL FLOW DIAGRAMS

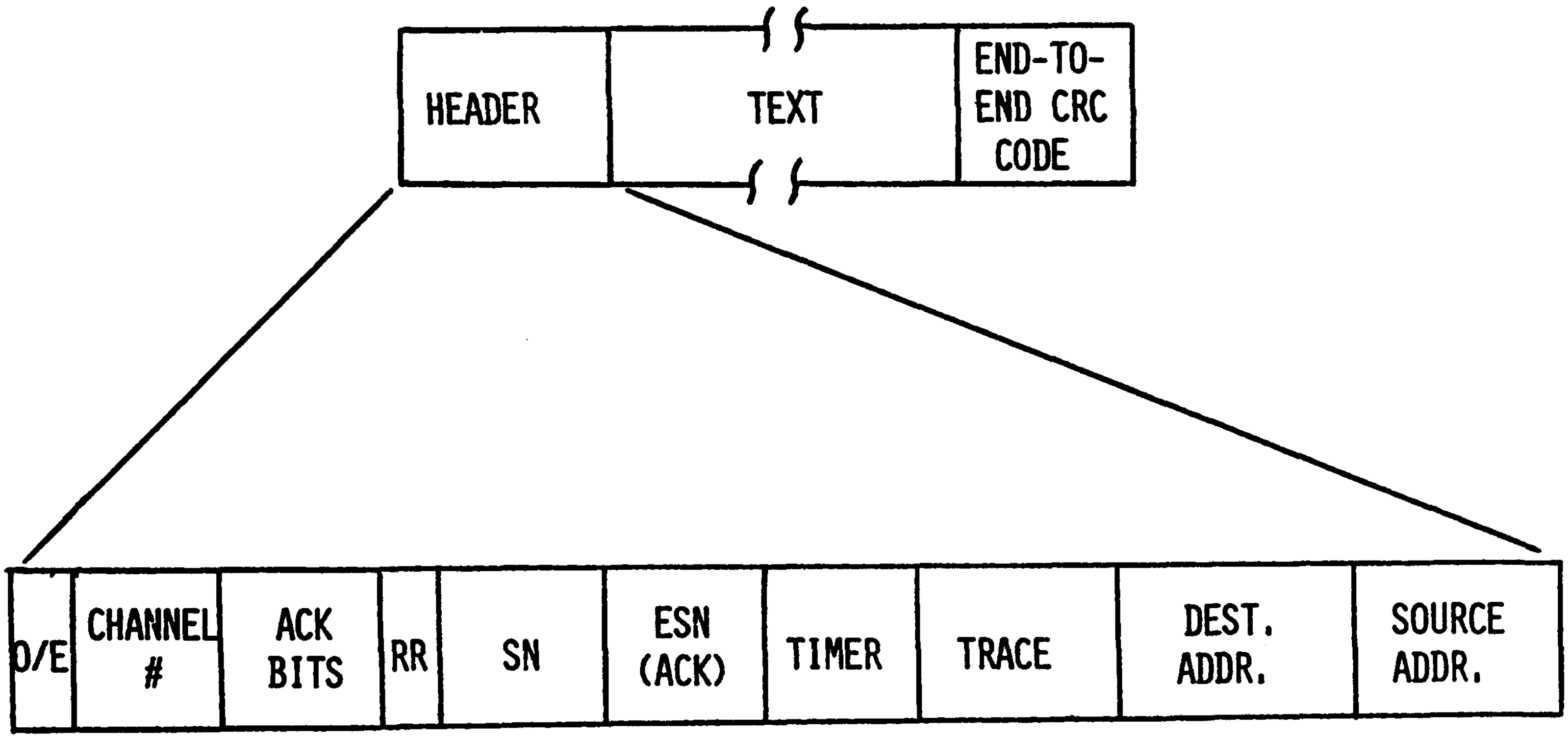


FIGURE 10. POTENTIAL CONTROL PACKET FORMAT

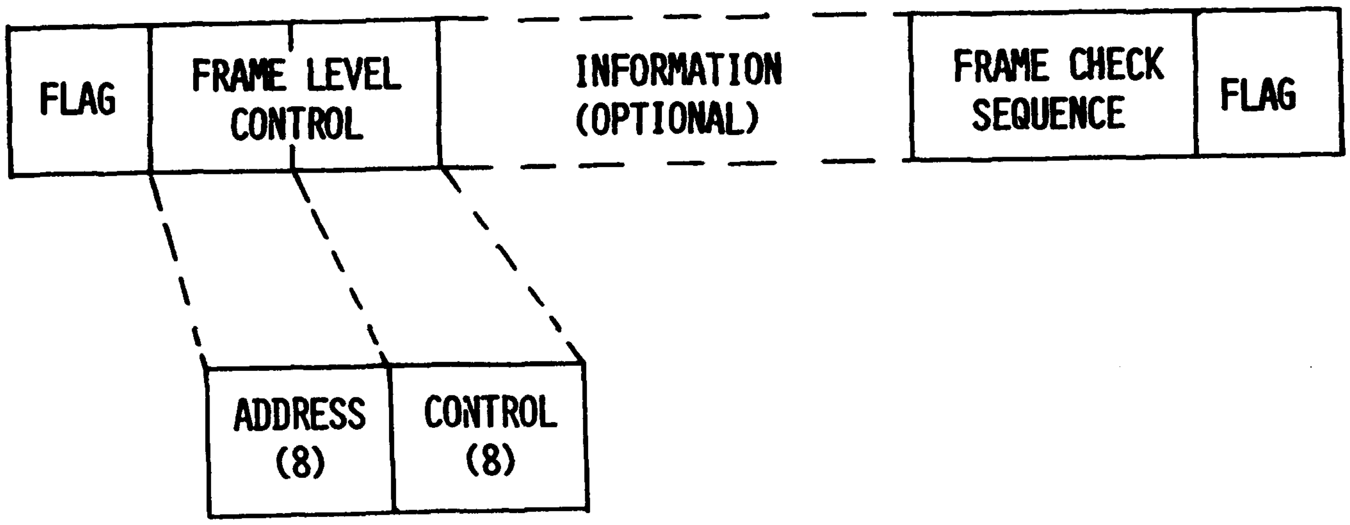


FIGURE 11. THE HDLC FRAME FORMAT

1	2	3	4	5	6	7	8	FRAME CLASS
0	N(S)			P/F	N(R)			INFORMATION
1	0	S		P/F	N(R)			SUPERVISORY
1	1	M		P/F	M			UNNUMBERED

N(S) = SEQUENCE NUMBER OF FRAME

N(R) = NEXT EXPECTED FRAME NUMBER

P/F = POLL/FINAL BIT

S = SUPERVISORY FRAME TYPE

M = UNNUMBERED FRAME TYPE

FIGURE 12. HDLC CONTROL FIELD

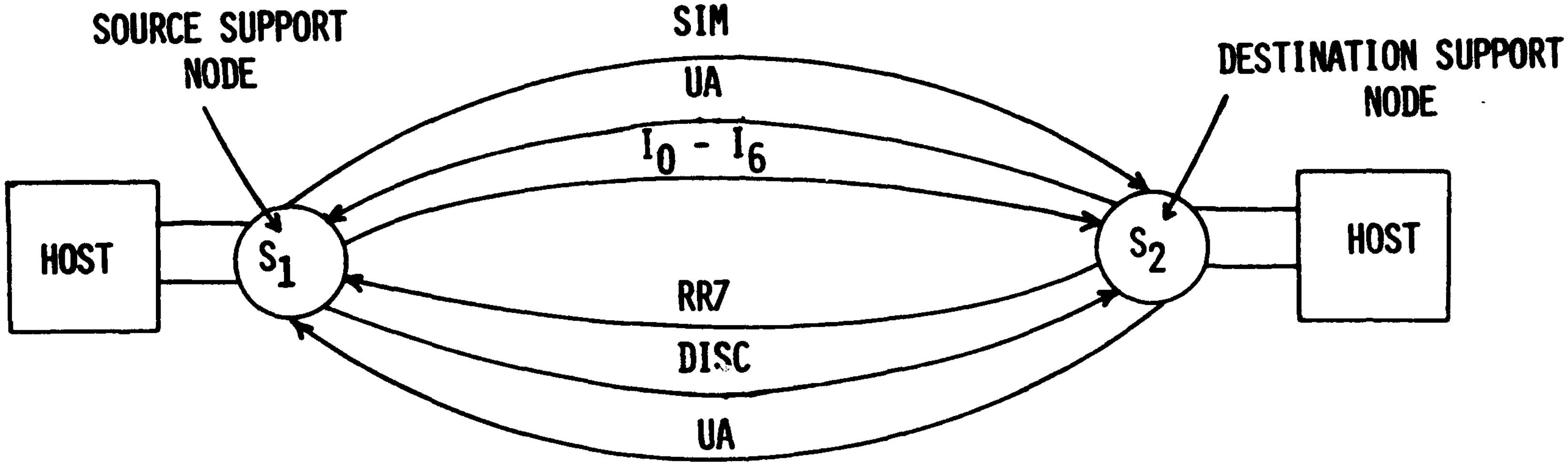


FIGURE 13. A HOST TO HOST DATA TRANSMISSION WITH HDLC PROCEDURES

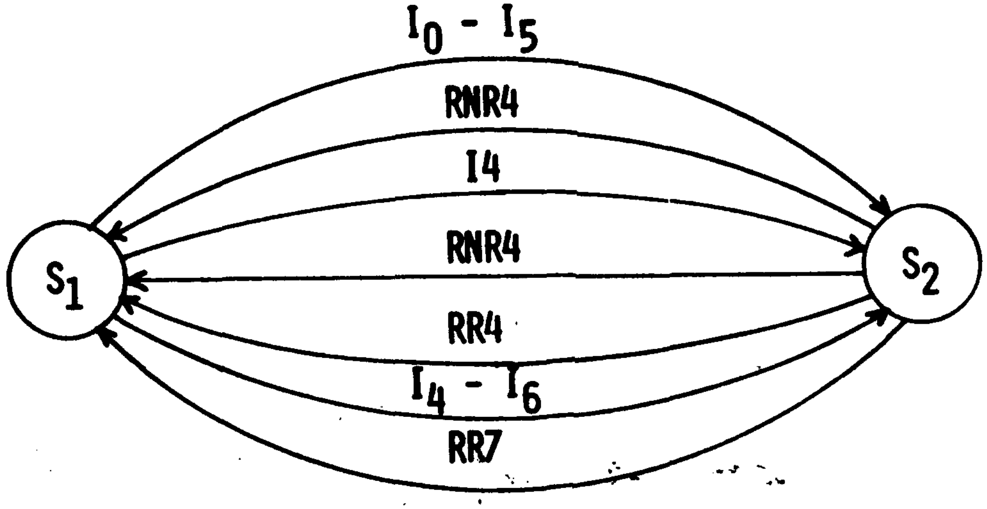


FIGURE 14. FLOW CONTROL EXAMPLE USING "RECEIVE NOT READY"

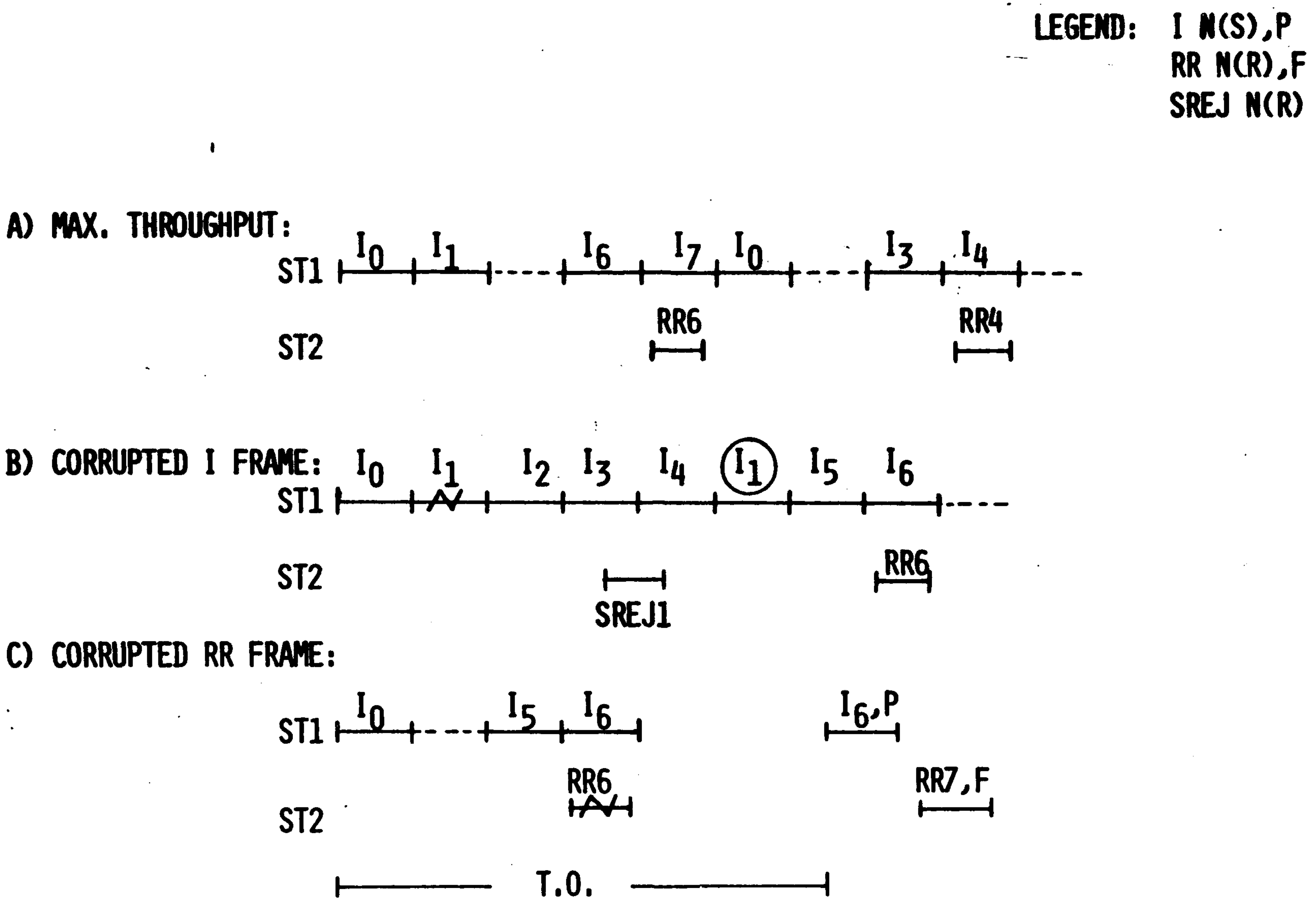


FIGURE 15. EXAMPLES OF LARGE DATA FILE TRANSFERS

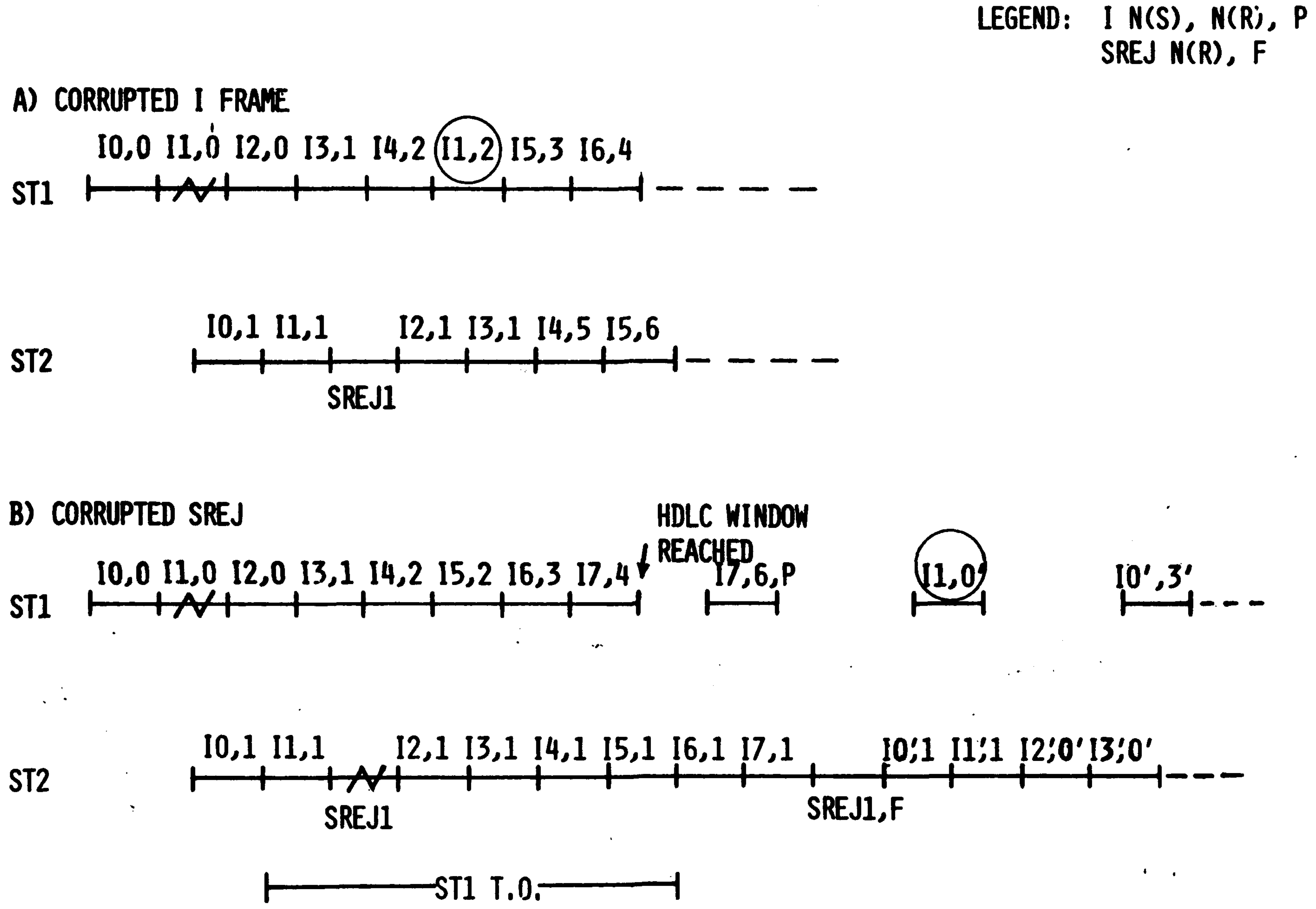

LEGEND: I N(S), N(R), P
SREJ N(R), F
A) CORRUPTED I FRAME
ST1
I0,0 I1,0 I2,0 I3,1 I4,2 I1,2 I5,3 I6,4
ST2
I0,1 I1,1 I2,1 I3,1 I4,5 I5,6
SREJ1
B) CORRUPTED SREJ
ST1
I0,0 I1,0 I2,0 I3,1 I4,2 I5,2 I6,3 I7,4
HDLC WINDOW REACHED
I7,6,P
I1,0
I0',3'
ST2
I0,1 I1,1 I2,1 I3,1 I4,1 I5,1 I6,1 I7,1
SREJ1
I0',1 I1',1 I2',0' I3',0'
SREJ1,F
ST1 T.O.

FIGURE 16. EXAMPLES OF INTERACTIVE HOST TO HOST COMMUNICATIONS

1. Report No. NASA TM-86304	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Communications Protocols for a Fault Tolerant, Integrated Local Area Network for Space Station Applications		5. Report Date September 1984
		6. Performing Organization Code 506-58-13-02
7. Author(s) Barry D. Meredith		8. Performing Organization Report No.
		10. Work Unit No.
9. Performing Organization Name and Address NASA Langley Research Center Hampton, VA 23665		11. Contract or Grant No.
		13. Type of Report and Period Covered Technical Memorandum
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

The evolutionary growth of the Space Station and the diverse activities onboard are expected to require a hierarchy of integrated, local area networks capable of supporting data, voice and video communications. In addition, fault tolerant network operation is necessary to protect communications between critical systems attached to the net and to relieve the valuable human resources onboard Space Station of day-to-day data system repair tasks. An experimental, local area network is being developed which will serve as a testbed for investigating candidate algorithms and technologies for a fault tolerant, integrated network. The establishment of a set of rules or protocols which govern communications on the net is essential to obtain orderly and reliable operation. A hierarchy of protocols for the experimental network is presented and procedures for data and control communications are described.

17. Key Words (Suggested by Author(s)) Protocols; Fault Tolerant, Integrated Network; Circuit Switching; Control Packets; Error Control; Experimental, Mesh Network; HDLC.	18. Distribution Statement Unclassified-Unlimited Subject Category 62

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 50	22. Price A02

For sale by the National Technical Information Service, Springfield, Virginia 22161